



VB.net

Second Term

3rd Prep

By

Ekhlas Fekry



www.Cryp2Day.com

موقع مذكرات جاهزة للطباعة

Chapter One

The Data



تطبيق مذكرات جاهزة للطباعة

تحميل من

App Store



احصل عليه من

Google Play

حمل التطبيق على موبايلك الأندرويد أو الأيفون

موقع مذكرات جاهزة للطباعة - www.cryp2day.com

Preface :

In the previously term we learned how to design a program's interface. Such as these application :

Note :

- ♦ The user entered data in the textboxes and chose other data, we note the values of these data are different types where;
 - The Name: represents a Text
 - Birthdate: represents a Date
 - Type: represents a Boolean value
 - The number of family: represents a numeric value
- ♦ These values are stored in the computer memory (RAM); to deal with these data we must assign a name and a type to it in memory, this type depending on the nature of the data element stored.

Data types :

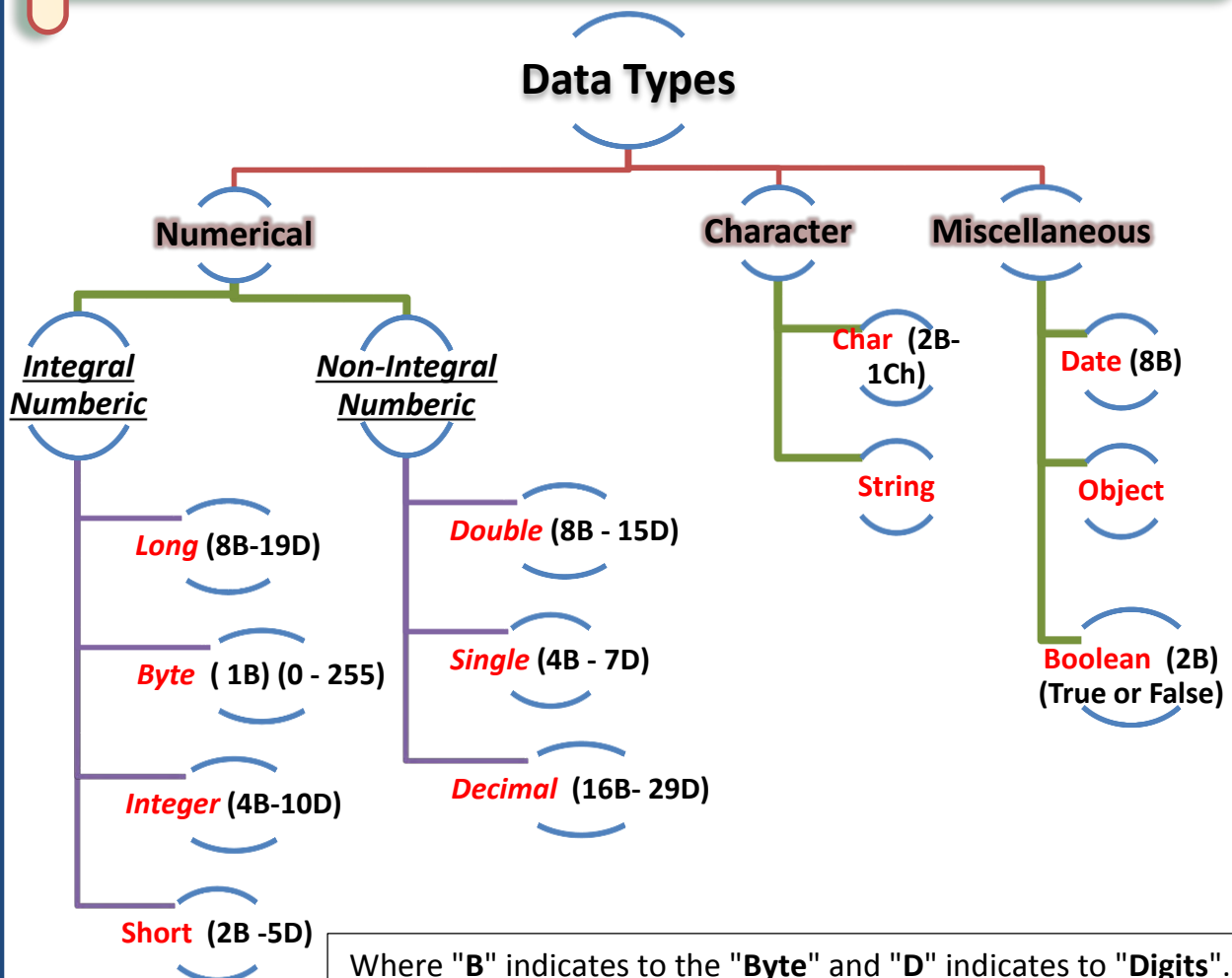
- ♦ In VB.net language, there are many Data Types.
- ♦ These data types are used to store input values from the user or resulting from execution of instructions of the program.

♦ Some Data Types provided by (Visual Basic.Net) are :

- 1) Numerical Data Types; to store Numerical Data :
 - a- Integral Numeric Types (**Byte** – **Long** – **Integer** – **Short**).
 - b- Non Integral (Fractional) Numeric Types (**Double** – **Single** – **Decimal**).
- 2) Character Data Types (**String** – **Char**).
- 3) Miscellaneous Data Types are those data that; do not fall under the Numeric or the Character types (**Object** – **Date** – **Boolean**).

NOTICE :

- Each classification of Data Type has more than one type.
- Each Data Type has:
 1. A storage space in the memory: *for example* the data type (**Integer**); when used, it occupies (يحتل) (4 bytes) .
 2. A range of values (minimum value and maximum value); *for example* the range of values for the data type (**Byte**) starts with '0' and ends with '255'.



Variables and Constant

The Variables :

- ◆ Variables are reserved places in computer memory (RAM) to store values temporarily, these values change during the running of program.
- ◆ Each Variable has a name and Data type.
- ◆ These Data Types are created by declaring variables then assigning values to them depending on their Data Types.

Declaration of Variables :

- ♣ We use "Dim" keyword to declare the variable.

Dim VariableName **As** Data type [= Initial Value]

Example :

Dim S_Name **As** String = "Ahmed"

Variable Name

Data Type

Initial Value will be change during the program running and it is **optional**

Dim F_Name **As** String

Code means: the declaration of a variable name as (F_Name) in memory, it's type is (String).

Dim Total_Price **As** Single

Code means: the declaration of variable its type (Single), name as (Total_Price) in memory.

Dim Today **As** Date = #1/25/2011#

(Complete)

Code meaning: the declaration of variable with the type....., name as (.....) in memory, and the initial value is (.....) for it.

The Constants :

- ① They are places reserved in the (RAM) and, have data types.
- ② You should be declaring the constants and assigning values of them.
- ③ These values are a fixed and cannot be changed during program execution.

Declaring Constants :

- ♣ We use the (**Const**) keyword to declare constants.

Const ConstantName **As** DataType = Value

Examples :

▪ **Const C_Name As String** = " جمهورية مصر العربية "

Meaning of code: the constant (C_Name) was declared, of data type (String) and, its text value " جمهورية مصر العربية " is assigned during the declaration

▪ **Const Pi As Single** = 22 / 7

Meaning of code: the constant (pi) was declared, of data type (Single) and, its numeric value 22/7 or 3.14 is assigned during the declaration.

▪ **Const BirthDate As Date** = #1/25/2011#

Complete:

The constant (-----) was declared in memory, of data type (-----) and, the value ----- is assigned during the declaration.

NOTICE :

- ♠ The double quotes " " are used if the value of variable or constant is a string value.
- ♠ The hashes # # are used if the value of variable or constant is date or time.

Naming rules Variables & Constants:

- 1) Variable or Constant names must begin with a letter or underscore (_).
- 2) Variable or Constant names should not contain symbols or special characters (e.g.: ?, *, ^, -, +, etc.) .
- 3) Variable or Constant names consist of letters, numbers, and underscores (_).
- 4) Do not use reserved words (Visual Basic.NET Language Keywords) *such as* (single, Dim, As).
- 5) It is preferable that the Variable or Constant name reflects its content.

Exercise (1-1) (Declaring Variables & Constants)

Prepare a program using VB.NET language that receives a numeric value for the radius of a circle, and its circumference and area is calculated when you press the "Command button".

Note that: area of a circle = πr^2 and the circumference of a circle = $2\pi r$ where: r represents the radius, $\pi = 22/7$

- Design the window Form.
- Then open the code window by pressing the button (F7) and, add the (Click) event procedures for (Button1, Button2).

Write the code required to calculate the Area of a circle using the Event Procedure (Button1_Click)

and, the circumference of a circle using the Event Procedure (Button2_Click)

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    End Sub
End Class
```

- In the scope of the event procedure (Button1_Click), type the code that calculates the **area** of a circle, and display the output in label control tool (Label2) when you click on (Button1) as shown in

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim Radius As Single
        Const Pi As Single = 22 / 7
        Radius = TextBox1.Text
        Me.Label2.Text = Pi * Radius ^ 2
    End Sub
End Class
```

Specify the scope of declaration for Variables and Constants : Exercise (1-2)

- In the scope of the event procedure (Button2_Click), type the code that calculates the **circumference** of a circle, and display the output in label control tool (Label2) when you click on (Button2) as shown in

But in event procedures for (Button2), the messages inside a yellow rectangle are displayed alerting (تنبيه) that the (Radius) and the constant (Pi) are *not declared*, although they are declared in the scope of the Event Procedure (Button1_Click).

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    R = TextBox1.Text
    Me.Label2.Text = Pi * R * 2
End Sub
```

Error Correction Options (Shift+Alt+F10)

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    R = TextBox1.Text
    Me.Label2.Text = Pi * R * 2
End Sub
```

Error Correction Options (Shift+Alt+F10)

So, to solve this problem we repeat the declaration of the variable (Radius) and the constant (Pi) in the scope of the Event Procedure (Button2_Click)

OR

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim Radius As Single
        Const Pi As Single = 22 / 7
        Radius = TextBox1.Text
        Me.Label2.Text = Pi * Radius ^ 2
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Dim Radius As Single
        Const Pi As Single = 22 / 7
        Radius = TextBox1.Text
        Me.Label2.Text = 2 * Pi * Radius
    End Sub
End Class
```

We declare the variable (Radius) and constant (pi) on the level of class (Form3) and thus we do not need to declare them on the level of the event procedure for each of the command button (Button1) or (Button2), as shown in figure

```

Public Class Form1
    Dim Radius As Single
    Const Pi As Single = 22 / 7
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Radius = TextBox1.Text
        Me.Label2.Text = Pi * Radius ^ 2
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Radius = TextBox1.Text
        Me.Label2.Text = 2 * Pi * Radius
    End Sub
End Class

```

When you run the program by pressing on F5 to do (Start Debugging), and enter a radius, we get the output, as shown in figure

NOTICE :

We can declare Variables and Constants on a Class level and then, we do not need the declaration process at each scope of the Event Procedure

Exercise (1-3) (Declaration of Variables)

Write VB.Net code necessary to declare the variables we need to receive the data that the user entered through a form "New user registration Form" by the following :

1) Design a form window as follows:

2) Use the following table to declare variables:

captions of Control tools	Data Type	variable names
اسم المستخدم	String	U_Name
تاريخ الميلاد	Date	U_B_D
النوع	Boolean	U_Gender
عدد أفراد الأسرة	Integer	U_C_F

3) Add the (Event Procedure), also called (Event Handler) for the button (التسجيل).by double-clicking on it.

4) Write the following code:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim U_Name As String
    Dim U_B_D As Date
    Dim U_Gender As Boolean
    Dim U_C_F As Integer
End Sub

```


Assignment Statement :

- ☞ It is a statement consists of two sides (right hand side and left hand side) separated by the assignment operator (=).
- ☞ It consists of taking the value on the right side of the assignment operator (=) and storing it in the element on the left.

Left side	Assignment operator	Right side
(Variable) <u>OR</u> (Property)	=	Abstract value <u>OR</u> Value from a (Variable) <u>OR</u> (Property) <u>OR</u> Value from an (Expression)

Assignment Examples:

Variables

- Number = 5 → *Abstract value*
- Number2 = Number → *Value from a (Variable)*
- Name = TextBox1.Text → *Property*
- Number = Number / 3 → *Value from an (Expression)*

Property → Label3.Text = "جمهورية مصر العربية" → *Abstract value*

NOTICE :

1. You can make assignment for (Variables) during the declaration or at a later stage. i.e. (Dim Number As Single = 5.6)
2. Reassignment of variables can be made many times, during the execution of the program.
3. The values assigned to variables can be (abstract values, Value from variables, property Or expression).
4. The assignment operator (=) doesn't indicate the arithmetic equality, it means "is replaced by" i.e. (x=x+1), where (x) in the left side represents the variable or storage, and (x) in the right side represents the value.

Exercise (1-4): Assigning values to Variables

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object,
        REM decleration of variables
        Dim U_Name As String
        Dim U_B_D As Date
        Dim U_Gender As Boolean
        Dim U_C_F As Integer

        REM Assigning values to the variables
        U_Name = TextBox1.Text
        U_B_D = TextBox2.Text
        If RadioButton1.Checked = True Then
            U_Gender = True
        End If
        If RadioButton1.Checked = False Then
            U_Gender = False
        End If
        U_C_F = TextBox3.Text
    End Sub
End Class
```

Assigning
values to
the
variables

Exercise (1-5): Using Variables:

To modify the form window to appear as follows, in order to display the values of variables in a label:

Add a control tool "Label5" as follows

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        REM decleration of variables
        Dim U_Name As String
        Dim U_B_D As Date
        Dim U_Gender As Boolean
        Dim U_C_F As Integer

        REM Assigning values to the variables
        U_Name = TextBox1.Text
        U_B_D = TextBox2.Text
        If RadioButton1.Checked = True Then
            U_Gender = True
        End If
        If RadioButton1.Checked = False Then
            U_Gender = False
        End If
        U_C_F = TextBox3.Text
        ' using variables
        Me.Label5.Text = U_Name & vbCrLf & U_B_D & vbCrLf & U_Gender & vbCrLf & U_C_F
    End Sub
End Class
```

Declaration of

Assignment values to

Using Variables

Express the current

To concatenation of texts

"vbCrLf" It used to create a new line.

(_) Enables writing lines of code in more than one line

NOTICE :

The previous code presents the Assignment statement of a set of variables for the property (Text) and the control (Label5) where:

1. Variables are separated from each other's by (& vbCrLf &)
2. The symbol (&) is used to Separates each variable and the other by the concatenation تتابع of texts.
3. The reserved word (vbCrLf) is used to create a new line.
4. Use the symbol (_) to write the lines of code in more than one line, if the code line is too long, to organize and simplify the process of reading code.
5. (Me) express the current (Form).
6. The programmer can use the command (Rem) in writing remarks that can be referred to within the code, it is not compiled.

Priority Rules of Arithmetic Operations (Operator Precedence) :

1. Applying the process inside the brackets from the inside to the outside.
2. Applying the exponent.
3. Applying multiplication or division process from left to right, wherever comes first.
4. Finally, the Application of the addition or subtraction process from left to right, wherever comes first.

The Error :

1 > Syntax Error : الأخطاء الإملائية والنحوية :

This happens, when writing code incorrectly.

Examples

- Din X As Single

The variable (X) was declared but there is a mistake in writing the word (Dim)

▪ **Const X As Single**

The constant (X) was declared but, its value is not assigned during the declaration.

✎ This type of error is easy to detect because, the (IDE) help us as it does not allow any error of this type.

2> Logic Error اخطاء منطقية:

It happens when we get incorrect results after executing the program because of the wrong formulating arithmetic or logic expressions

Example:

Write the code => `label2.text = Pi + Radius ^2` instead of

The code => `label2.text = Pi * Radius ^2` to calculate the Area of a Circle.

✎ To overcome this type of error, you must review the written code, and test the program with data already validated.

3> Runtime Error :

- These errors are discovered when the program is running..
- It is found in lines of code, where the Assignment Statement is written.

Example:

When you assign a value greater than or less than the range value of Data Type used **OR** assign a value that is not equivalent (لا تعادل) to the variable's Data Type .

Questions

First: put a Tick (✓) in front of the correct statement and a cross (✗) in front of the wrong statement for each of the following sentences:

NO.	Question	Answer
1	One of the advantages of VB.NET is dealing with different types of data.	()
2	One of disadvantage of VB.NET is dealing with different types of data.	()
3	All the data entered into the VB.NET program language are stored temporarily in the computer memory.	()
4	All types of data saved in the memory occupy the same storage space.	()
5	A good programmer is the one who improves the rationalization of storage space in the computer memory.	()
6	The value of the student's total grades is classified within the integer data types.	()
7	The value of the student's name is classified in the Miscellaneous data types.	()
8	The value of the student gender "male" or "female" is classified within the Miscellaneous data types " Boolean ".	()
9	Image of a student can be classified within the character data types.	()
10	The value of the employee's salary can be classified within non-integer numeric data types.	()
11	Each data element stored in computer memory occupies a particular storage space and a particular range of values according to its data type.	()
12	The data element identifies the storage space it occupies in computer memory and knowing the minimum and the maximum for its value.	()
13	The term variables in vb.net means stores in the computer memory, which has type and name.	()
14	Declaring a variable in VB.NET means determining its name and data type.	()
15	The declaration of variables in the language VB.NET helps rationalize the use of the computer memory.	()
16	Declaration of variables is a matter of formality, because VB.NET languages recognize the variables and determine the type automatically.	()
17	The following statement " Dim F_name As String " is to declare the name of a variable " String " and type " F_name ".	()
18	The following data element " Dim F_name As String " is to declare the name of a variable " F_name " and type " String ".	()
19	The declaration statement for variables is determined by the variable name and type.	()
20	The declaration statement for the variables is determined by the name, type and fixed value.	()
21	"55City" variable name is a consider a wrong variable name because it begins with a number.	()

NO.	Question	Answer
22	"55City" is considered a variable correct name.	()
23	"Name" is considered a correct variable name in event procedure level (enrichment).	()
24	"Name" is considered a correct variable name in form1class level (enrichment) اثرائى	()
25	"Dim" is used to declare variables.	()
26	"Dim" is used to declare constants.	()
27	The command "Const" is used in the declaration of the variables.	()
28	The command "Const" is used in the declaration of the constants.	()
29	Constants in VB.NET language are stores of a computer memory which have the name and the value that does not change during the running of the program.	()
30	Constants in VB.NET language are stores of a computer memory which have name and value can change during the running of the program.	()
31	The error in the result of any equation is a Syntax Error.	()
32	The error in the result of any equation is a Logical Error.	()
33	Error that appears while you run or execute a VB.NET program is called Syntax Error.	()
34	Error that appears during the execution of VB.NET program is Run time Error.	()
35	The final value of the variable X after the execution the following equation "X = 3 + 2 * 4" is (11).	()
36	The final value of the variable X after execution the following equation "X = 3 + 2 * 4" is (20).	()

Second: Select the appropriate answer to complete each of the following sentences:

- (1) The value of prices of desktop tools can be classified as..... data.
a) integer b) non- integer c) miscellaneous
- (2) The value of the names of the subjects can be classified as data.
a) miscellaneous b) non- integer c) string
- (3) The type of data element temporarily stored in the computer memory defines:
a) storage space and the extent of its value
b) name and storage space
c) storage space and a storage value
- (4) The right syntax to declare Salary variable is
a) Dim Salary As Integer b) Dim Salary As Byte c) Dim Salary As Decimal
- (5) The right syntax to declare the city variable is
a) Dim City As String b) Dim City As Byte c) Dim City As Decimal
- (6) The right syntax to declare the variable name F_Name is
a) Dim F_Name As Integer b) Dim F_Name As String c) Dim F_Name As Decimal
- (7) The right syntax to declare the variable Gender is
a) Dim Gender As Decimal b) Dim Gender As Integer c) Dim Gender As Boolean
- (8) The right syntax to declare the variable name F_Name is
a) Din F_Name As String b) Dim F_Name As String c) Dim F_Name As Char

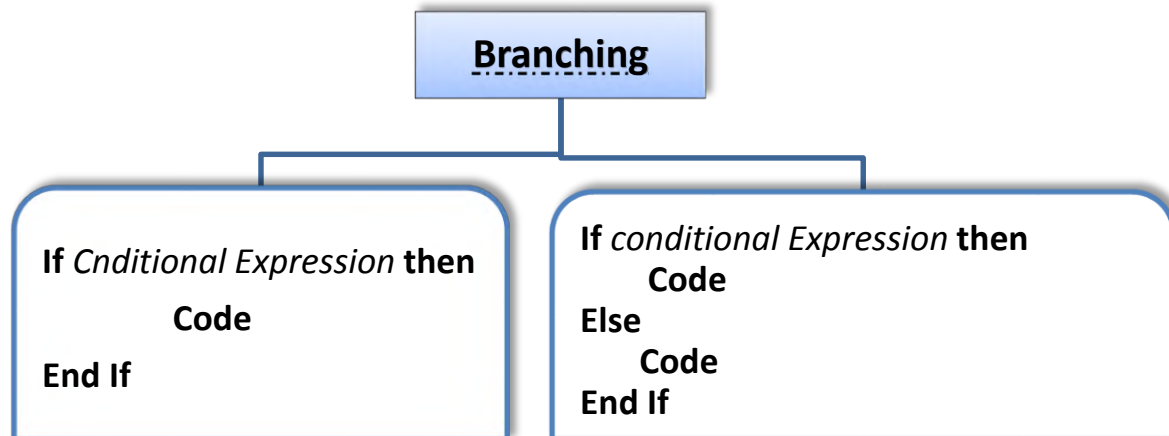
- (9) The error that appears after running a program VB.NET language is called
 a) Syntax Error b) logical Error c) Runtime Error
- (10) The error that appears while writing a code in a VB.NET language called.....
 a) Syntax Error b) Logical Error c) Run time Error
- (11) The error in the output result in language VB.NET code is called.....
 a) Syntax Error b) Logical Error c) Run time Error
- (12) The final output of the variable X for equation " $X = 3 + 2 * 4$ " is
 a) 11 b) 24 c) 20
- (13) The final output of the variable Y for the equation " $Y = 16 - 12/4 + 2$ " is
 a) 3 b) 11 c) 15
- (14) The declaration statement of a variable "Dim X As String", means the declaration about
 a) A variable named X and type of character String.
 b) Variable called string and its type X.
 c) Unknown variable has no name and its type String.
- (15) The correct statement to declare a non-integer variable named Y is
 a) Dim Y As Decimal b) Y As Decimal c) Dim Y = Decimal
- (16) Choose the correct name of the variable "name of the student":
 a) st_name b) st name c) Name**
- (17) Choose the correct name of the variable "address of the employee":
 a) 5Cairo b) E_Address c) (Address)
- (18) The names of the following variables are correct in level of class form1 except: (enrichment)
 a) St_text b) Text c) _st_text
- (19) On declaration of a mathematical constant π , we use the code
 a) Dim Pi As Single b) Dim Pi As Single = 3.14 c) Const Pi As Single = 3.14
- (20) On the declaration of constant gravity acceleration, we use the code
 a) Dim g As Single b) Const g As Single = 9.81 c) Dim g As Single = 9.81
- (21) The declaration of variable number of family members C_Family with initial value of 2 is:
 a) Dim C_Family As Single = 2
 b) Const C_Family As Integer = 2
 c) Dim C_Family As Integer = 2
- (22) If there is an error in the result of a rectangular area calculation in a program, this error is considered
 a) Syntax Error b) Logical Error c) Run time Error
- (23) The error message that appears when you write the code "Dimension X As Byte", can be classified as
 a) Syntax Error b) Logical Error c) Run time Error
- (24) The final output of the equation " $Y = 12 - 2 + 4 / 2$ " is
 a) 12 b) 7 c) 9
- (25) The final output of the equation " $Y = 12 - (2 + 4) / 2$ " is
 a) 12 b) 7 c) 9

Chapter Two

Branching

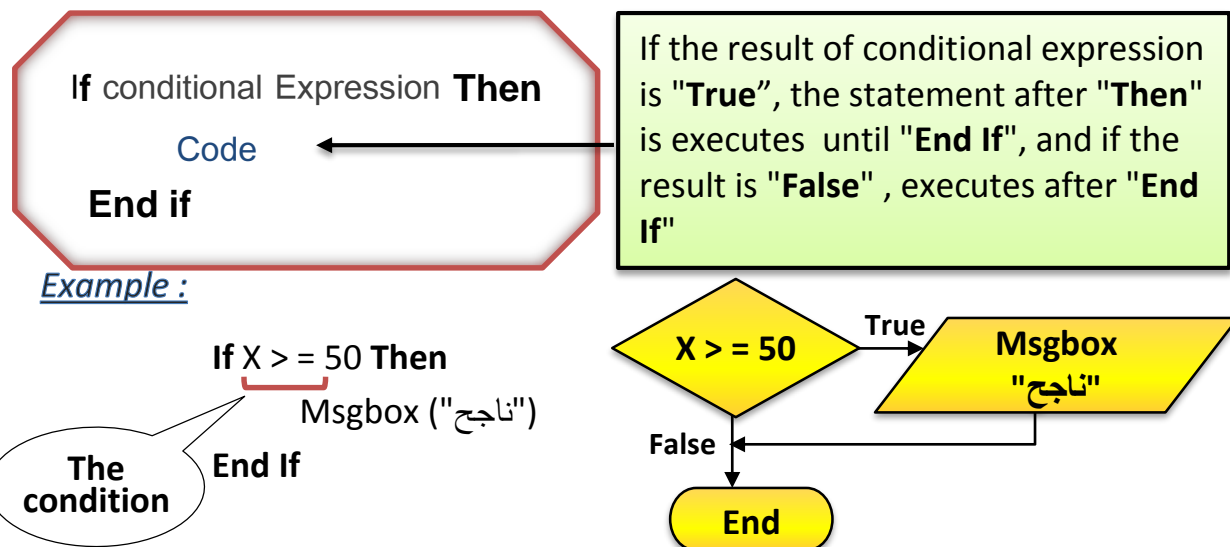
Preface :

In this chapter, we will learn the branching; where it is used to execute a sequence of steps depending on the result of the condition or the answer of the question. You will find that writing code is just translating an algorithm into a programming language.



Branching Statement using (If Then) :

The syntax of (If...Then) statement



In this example, if the result of condition is (True) it displays message box "ناجح" then executes the statement after (End If), but if the result is (False), it executes the statement after (End If).

The Conditional Expressions:

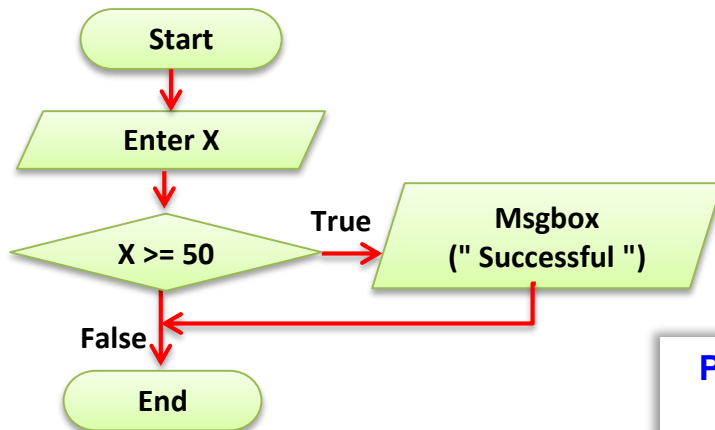
- A conditional expression is part of a program code; its result can be (True) or (False) depending on the value of: (a Property or a Variable or another piece of data in the program code).

- It consists of **three parts** : logical operator preceded by (مسبوق بـ) an abstract value ,a value of a variable or constant or a result of a mathematical expression that is compared with an abstract value ,a value of a variable or constant or a result of a mathematical expression,

Conditional Expression			Example of conditional expression
Before logical operator	logical operator	After logical operator	
Variable	> (greater than) < (Less than)	Abstracted value	If A > 5 If A < 5
Abstracted value	<> (not equal)	Variable	If 5 < > A
Variable or constant	<= (less than or equal) >= (greater than or equal)	Variable or constant	If B <= A If B >= A
a value from expression	= (equal to)	a value from expression	If A^2 = B/c
Variable	= (equal to)	a value from expression	If B = A + 3 * 2
Property	= (equal to)	Logical value	If RadioButton1.checked = True
Property	<> (not equal)	Property	If TextBox1.text < > TextBox2.Text

Exercise (2-1):

Write the steps to create a program in which we enter student's score, then the message "Successful" appears in a message box if the score is greater than or equal to 50, when you click on the button "Result"



This (If) statement can be written, in one line without writing (End if)

```

Private Sub Button1_Click
    Dim x As Single
    x = TextBox1.Text
  
```

```

    If x >= 50 Then MsgBox("Successful ")
  
```

```

End Sub
  
```

```

Private Sub Button1_Click
  
```

```

    Dim x As Single
  
```

```

    x = TextBox1.Text
  
```

```

    If x >= 50 Then
  
```

```

        MsgBox("Successful ")
  
```

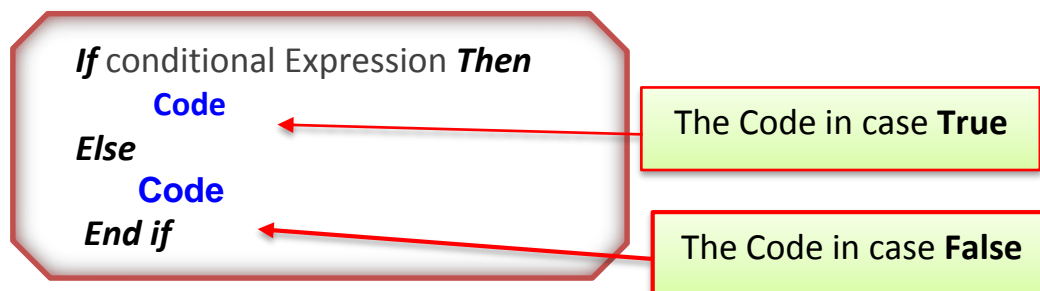
```

    End If
  
```

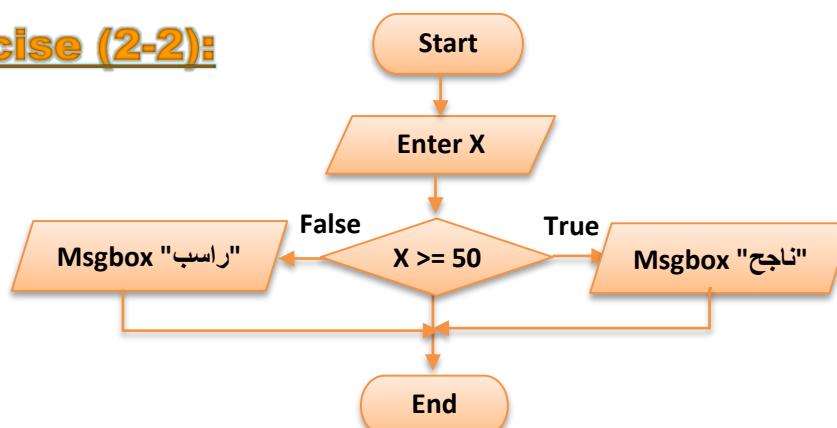
```

End Sub
  
```

Branching Statement using (If Then Else) :



Exercise (2-2):



Note:

This (If) statement can be written, in one line without writing (End if)

Private Sub Button1_Click

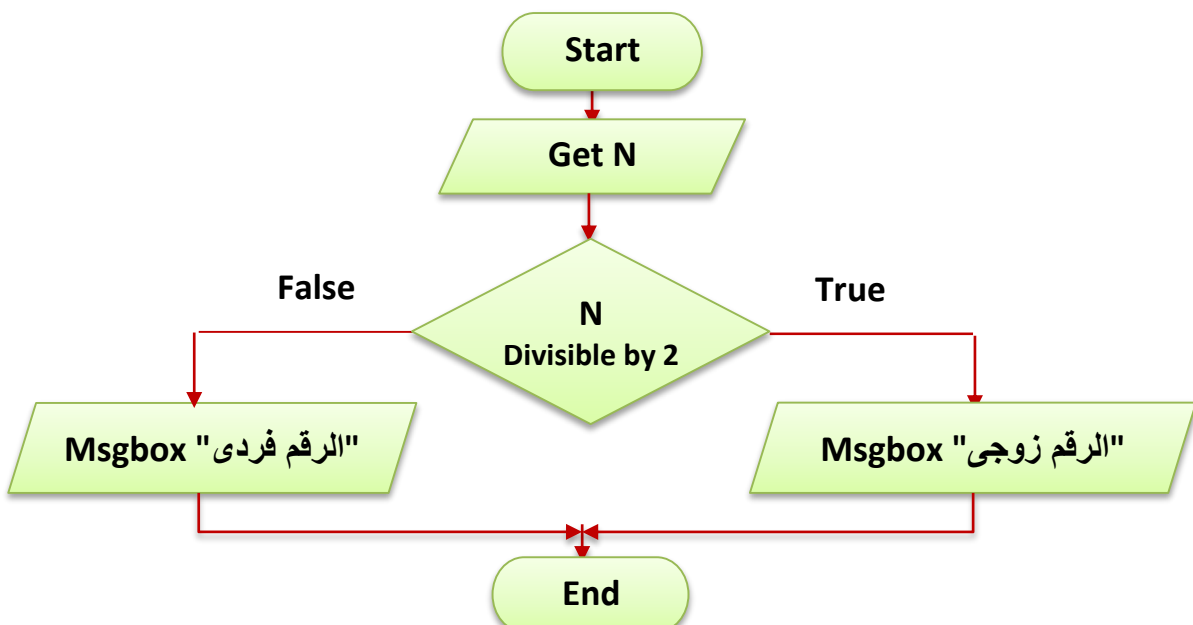
```
Dim x As Single
x = TextBox1.Text
If x >= 50 Then
    MsgBox ("ناجح")
Else
    MsgBox ("راسب")
End If
End Sub
```

Private Sub Button1_Click

```
Dim x As Single
x = TextBox1.Text
If x >= 50 Then MsgBox ("ناجح") Else MsgBox ("راسب")
End Sub
```

Exercise (2-3):

Create a new project that receives inputs in the variable (N) through a textbox, and displays the message "الرقم زوجي" or "الرقم فردي" in a MessageBox.



```

Private Sub Button1_Click
    Dim N As Single
    N = TextBox1.Text
    If N Mod 2 = 0 Then
        MsgBox ("الرقم زوجي")
    Else
        MsgBox ("الرقم فردي")
    End If
End Sub

```

Note :

Where (**Mod**) is a mathematical operator that computes the remainder of a division expression; when the remainder of a division by 2 equals zero, this means there is no remainder; therefore it is an even number.

Branching Statement using (Select Case) :

The (**Select ...Case**) statement is used in the branching depends only on the value of one variable and there are many conditions, which reduces the code and makes it easier and clearer

The syntax of (Select ...Case) statement :

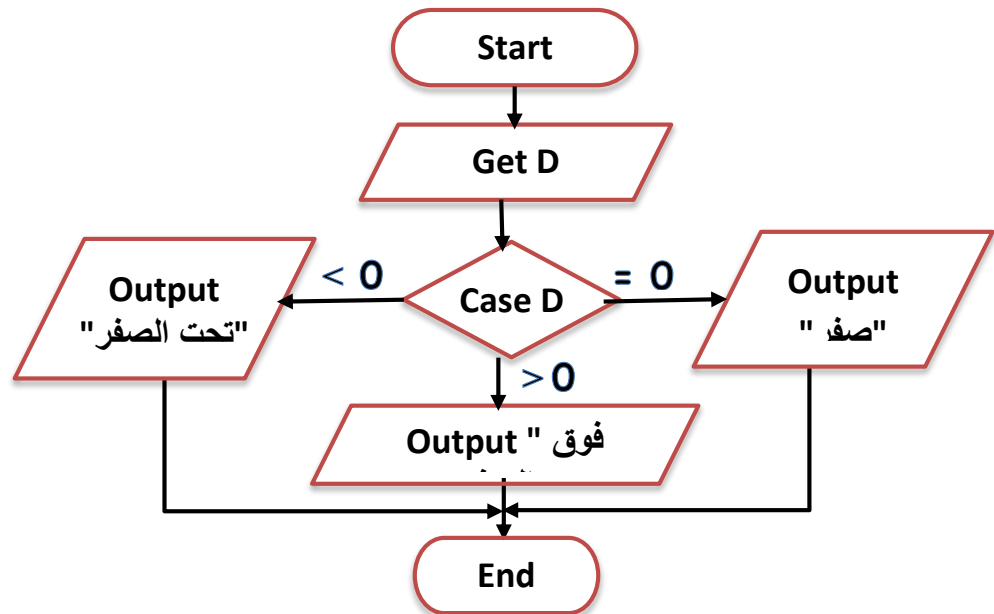
```

Select ...Case Variable
    Case value1
        Code
    Case value2
        Code
    Case value3
        Code
    ..
    ..
    Case else
        Code
End Select

```

Exercise (2-4)

on entering the temperture through the TextBox (TextBox1) and then clicking the "Test" button the phrase "above zero" OR "below zero" OR "equal zero" is displayed through (Label2). Try to draw a flowchart compatible with (Select Case)



Private Sub Button1_Click(**ByVal** sender **As** System.

Dim degree **As** Single

Try

degree = TextBox1.Text

Select Case degree

Case 0

Label2.Text = "درجة الحرارة تساوى صفر"

Case Is < 0

Label2.Text = "درجة الحرارة تحت الصفر"

Case Is > 0

Label2.Text = "درجة الحرارة فوق الصفر"

End Select

Catch ex **As** Exception

Label2.Text = ""

MsgBox("أدخل عدد فقط")

TextBox1.Text = ""

TextBox1.Focus()

End Try

End Sub

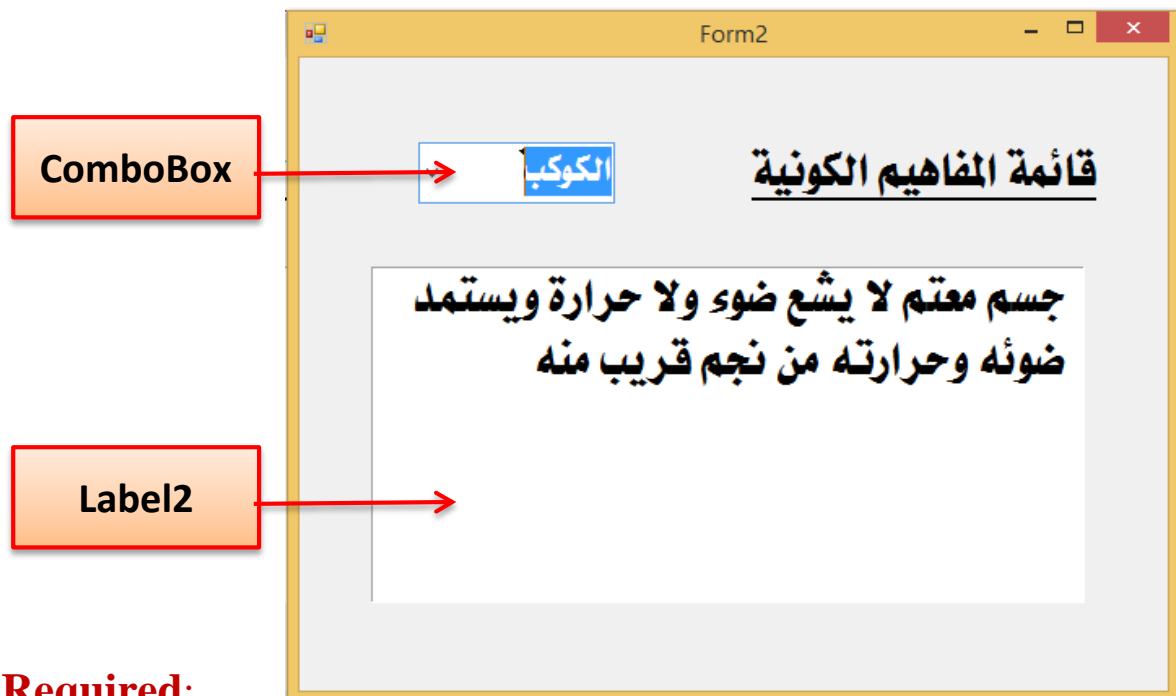
Note :

- The previous flowchart, illustrates more than two possible paths (branches) out of the decision point.
- This is a comparison that evaluates a question about the value of variable (D), and according to this value the branching of code differs.

Exercise (2-5)

In geography, you have learned many concepts on Galaxies, planets and stars ... etc.

It is required that when selecting one of the elements of a **ComboBox**, the corresponding definition appears in the **label**.

**Required:**

Type the necessary code, so that on selecting one of the elements of a ComboBox, its definition appears in the **Label2**.

Instructions

1. Set the value of the property (AutoSize) for the (Label2) to "False".
2. Add the following items (Galaxy - Planet – Moon – Star - Meteoroid - النيزك - falling star) for the (ComboBox) control through the property (Items).
3. Type the code in the (SelectedIndexChanged) event procedure for the control (ComboBox) as follows:

```

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Select Case ComboBox1.SelectedIndex
        Case 0
            Label12.Text = "تجمعات كبيرة من النجوم المختلفة الاحجام والاشكال والانواع"
        Case 1
            Label12.Text = "جسم معتم لا يشع ضوء ولا حرارة ويستمد ضوئه وحرارته من نجم قريب منه"
        Case 2
            Label12.Text = "جسم معتم أصغر حجما من الكوكب يرتبط به بفعل الجاذبية ويعكس ضوء الشمس الساقط عليه"
        Case 3
            Label12.Text = "جسم مضئ ملتهب يشع ضوءا وحرارة"
        Case 4
            Label12.Text = "جسم صلب كبير يحترق جزئيا عند احتكاكه بالغلاف الجوى"
        Case 5
            Label12.Text = "جسم سماوى صلب يسبح فى الفضاء حول الشمس ويجرى فى السماء على هيئة سهام ضوئية"
        Case Else
            MsgBox("يرجى اختيار احد المفاهيم")
    End Select
End Sub

```

Notice:

We have depended, in writing of the code, on the order of the elements within the tool (ComboBox) where the first item has an (index) 0, the second item has an (index) 1 and the third item has an (index) 2,and, so on.



Questions

- (1) Answer the questions with the help of the following code:

```

If X >= 50 Then
    MsgBox ("successful ")
End if
```

- A) Message box is shown with the text "successful" when :

.....

- A) If the value of X equals 50, the result of executing code is

.....

- B) If the value of X equals 62, the result of executing code is

.....

- (2) Answer the following questions with the help of the general syntax of the "IF Then" statement:

If (conditional expression) Then Cod 1 Else Code 2

- A) Write conditional expression testing the value of the variable Y if it is less than 0.


.....

- B) Replace the "Code 1" in the general syntax with a code that shows the text "a negative number" in the message box.

.....

- C) From the general syntax of statement (If .. Then .. Else) that if the conditional expression is "True", then is executed and if the expression is not achieved, then..... is executed . (Complete)

(3) Answer the following questions with the help of the screen and the code in the table:

Code	screen
<pre>Private Sub Button1_Click Dim x As Single x = Me.TextBox1.Text If x >= 50 Then MsgBox ("ناجح") End If End Sub</pre>	

- A) The purpose of the program is:
- B) The code is executed if the event occurred on control tool.
- C) Type of variable X in the code is
- D) "Me." In the code refers to
- E) We input the value (50) in the text box, the result of the implementation of the code is:

(4) Complete the following table with the required code, using the general syntax conditional statement "IF .. Then .. Else"

If Conditional Expression Then
Code 1
Else
Code 2
End if

So as to show a message box having the word "مصر" if the value of the variable "Country" is equal to the "Egypt" or message box having the word "Egypt" appears:

No	statement	code
1	Conditional expression
2	Result of achieving the condition "True"
3	Result of not achieving the condition "False"

(5) Answer the following questions with the help of the code:

F) If the value of X = 76, the result of executing the code is:

.....

G) If the value of X = 49, the result of executing Code is :

.....

H) Rewrite the code of "Block If" to appear on only one line.

.....

```
Dim x As Single
x = Me.TextBox1.Text
If x >= 50 Then
    MsgBox("ناجح")
Else
    MsgBox("راسب")
End If
```

(6) After studying the code, answer the following questions:

A) Modify the code so that the "الرقم الزوجي" text appears in a label "Label2" and "الرقم فردي" text appears in a label "Label2" instead of the message box.

```
Dim N As Long
N = Me.TextBox1.Text
If N Mod 2 = 0 Then
    MsgBox("الرقم زوجي")
Else
    MsgBox("الرقم فردي")
End If
```

B) Replace the type of variable "N" to be "Integer".

.....

.....

.....

.....

.....

.....

(7) The following code receives any number of a TextBox, and stores it in a variable, and then tests its value. If the number is even or odd, a MessageBox appears showing that.

Required: Retype the code after discovering the errors and correcting them so that the result of its implementation is right.

```
Dim X As Integer
N = Me.TextBox1.Text
If N Mod 2 = 0
    MsgBox("الرقم زوجي")
Else
    MsgBox ("الرقم فردي")
```

(8) After studying the following code, answer the following questions:


```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim degree As Single
    Try
        degree = Me.TextBox1.Text
        Select Case degree
            Case 0
                Me.Label2.Text = "صفر"
            Case Is < 0
                Me.Label2.Text = "تحت الصفر"
            Case Is > 0
                Me.Label2.Text = "فوق الصفر"
        End Select
    Catch ex As Exception
        MsgBox("ادخل عدد")
        Me.TextBox1.Focus()
        Me.TextBox1.Text = ""
    End Try
End Sub

```

- A) The purpose of the code is
- B) If you learn that: " Degree = -3" the text appears in the message box is :.....
- C) The code is executed when the event..... occurs on control tool..
- D) Type of variable "Degree" :is

Oh My God



Chapter Three

Looping & Procedures



Preface:

In this chapter, you will learn how to repeat a set of statements many times (which is called Looping) using (For... Next) and (Do... loop

Using (ForNext) statement:

It is one of the limited loop statements used when we want to repeat a code for specific number of time.

The syntax of the (For...Next) statement

For Variable = Start Value To End Value [Step Add Value]

VB code

Next [Variable]

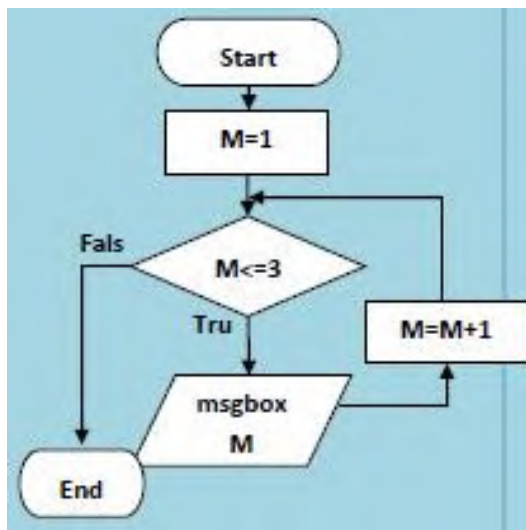
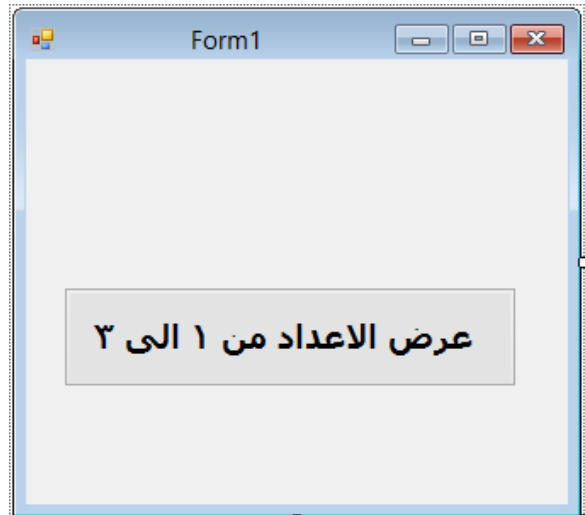
Counter
Variable

Numeric values

- The (For...Next) statement should start with the Keyword (For) and end with the Keyword (Next).
- "Variable" is the name which represents the counter and its type must be numeric (integer or decimal).
- "Start Value" is the start value of the counter or the beginning of repetition is a numeric value.
- "End Value" is the value of the end of the counter and the end of the repetition is also a numeric value.
- "Add Value" is the increment value of the counter or value over the counter until it reaches the end value.
- "Code" is one or more commands to be replicated between the beginning of loop (For) and its end (Next).
- The statements (VB code) are repeated inside the loop until the **counter value reaches the End value.**
- If the value of the increment is positive 1, it can be dispensed with writing (Step AddValue) as the default value to increase the counter positive 1.
- Typing a variable name counter next to "Next" optional.

Exercise (3-1)

Design the following form window, to have the numbers from 1 to 3 displayed in the MessageBox on pressing on the button "View numbers from 1 to 3".



```
Private Sub Button1_Click(By
```

```
Dim M As Integer
```

```
For M = 1 To 3
```

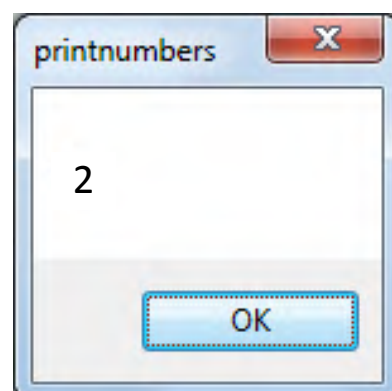
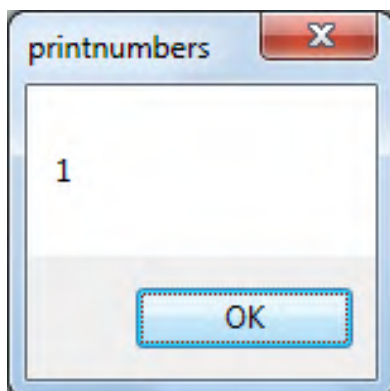
```
MsgBox(M)
```

```
Next
```

```
End Sub
```

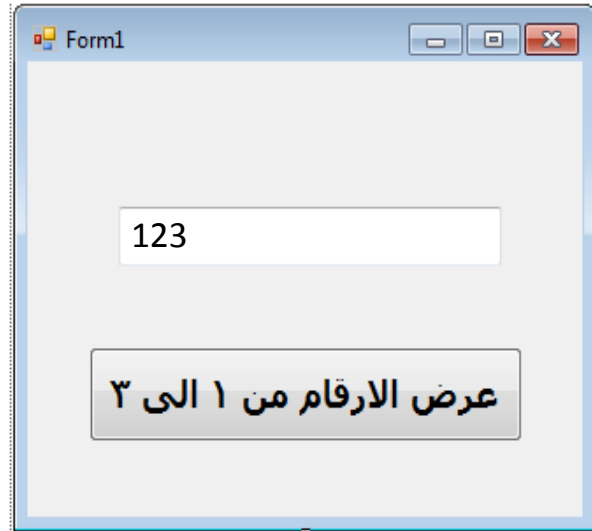
NOTICE:

We receive a message box that displaying the number "1" and when we click on the button "Ok" another message box appears displaying the number "2" and so on till the value 3.



Exercise (3-2)

Modify the code in the previous exercise to display the numbers from 1 to 3 in the (Textbox) :



```
Dim m As Integer
```

```
Me.TextBox1.Text = ""
```

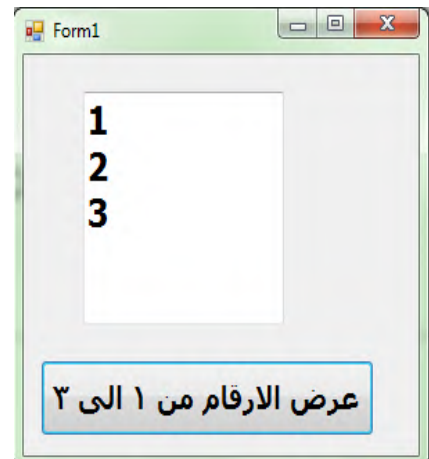
```
For m = 1 To 3
```

```
    TextBox1.Text = Me.TextBox1.Text & m
```

```
Next
```

This code to clear the content of textbox

To display each number in a new line :



This code to delete the content of textbox

```
Dim m As Integer
```

```
Me.TextBox1.Text = ""
```

```
For m = 1 To 3
```

```
    → TextBox1.Text = Me.TextBox1.Text & m & vbCrLf
```

```
Next
```

String constant used to add: the (enter key) symbol and a new line

VbCrLf = Visual Basic Carriage Return Line Feed

Exercise (3-3) :

Print the multiplication table of 3 in a (TextBox):

Dim M, product As Integer

Dim str As String

Me.TextBox1.Text = ""

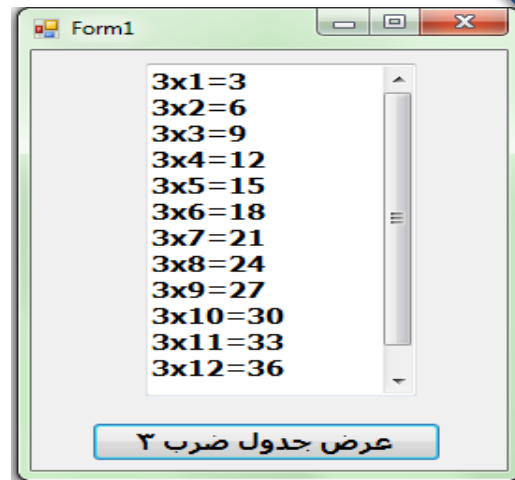
For M = 1 To 12

str = 3 & "x" & M & "="

Product = 3 * M

Me.TextBox1.Text = TextBox1.Text & str & Product & vbCrLf

Next M

**Notice:**

- ♣ The variable (product) is declared; to store the multiplication product; at each time the variable (M) changes its value.
- ♣ The variable name (str) is declared; to store the multiplication sentence (3x1= or 3x2=etc) as a string, at each time the variable (M) changes its value.
- ♣ You can write the code without using these variables as follows:

Dim M As Integer

Me.TextBox1.Text = ""

For M = 1 To 12

Me.TextBox1.Text = TextBox1.Text & 3 & "x" & M & "=" & 3 * M & vbCrLf

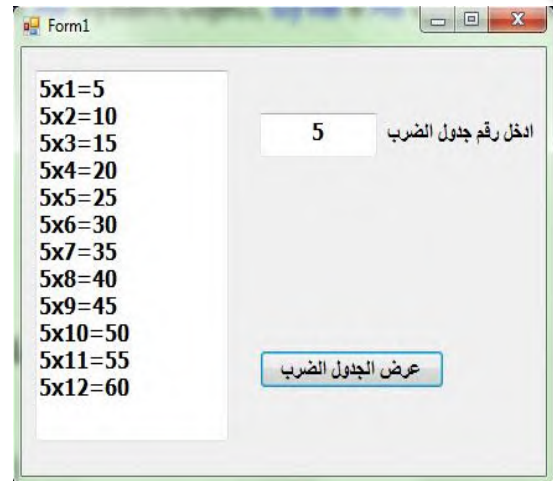
Next m

The multiplication
sentence

The multiplication
product

Exercise (3-4) :

Print the multiplication table of any number you enter in the (TextBox):



```
Dim M, product, NUM As Integer
```

```
Dim str As String
```

```
NUM = Me.TextBox2.Text
```

```
Me.TextBox1.Text = ""
```

```
For M = 1 To 12
```

```
    str = NUM & "x" & M & "="
```

```
    Product = NUM * M
```

```
    Me.TextBox1.Text = TextBox1.Text & str & Product & vbCrLf
```

```
Next M
```

Control the start value . end value and Step – increment in (For Next) statement

Examples :

S.	Example	Code
1	To display the odd numbers from 1 to 10	<pre>For I =1 To 10 Step 2 TextBox1.Text = TextBox1.text & I & vbCrLf Next</pre>
2	To display the even numbers from 2 to 10	<pre>For I = 2 To 10 Step 2 TextBox1.Text = TextBox1.text & I & vbCrLf Next</pre>

3	To display the numbers divisible (can be divided) by 3 from 3 to 20	For I = 3 To 20 Step 3 TextBox1.Text = TextBox1.text &I & vbCrLf Next
4	To display the even numbers descending order from 10 to 1	For I = 10 To 1 Step -2 TextBox1.Text = TextBox1.text &I & vbCrLf Next
5	To display the numbers from 1.50 to 0.5 with decremented by 0.05 each time	For I =1.5 To 0.5 Step -0.05 TextBox1.Text = TextBox1.text &I & vbCrLf Next
6	To display the numbers from 1 to the value of B at increasing value of C	For I = 1 To B Step C TextBox1.Text = TextBox1.text &I & vbCrLf Next

(Do WhileLoop) statement:

The syntax of the (Do While.....Loop)

Do While Conditional Expression

VB code

Loop

- ❖ The (Do While...Loop) is used to repeat a specific code for a several times of an unknown end, but based on a specific condition, so it is useful if you do not know the number of iterations .
- ❖ The code between the beginning of the loop "Do While" and its end will be implemented as long as the conditional expression is true. If the condition is not met for any reason, we get out of the iterative loop, and implement the code after the Loop if it exists.

Exercise (3-5) :

Design a project to enter a number in the TextBox , when click the button " أعداد فردية " the Odd numbers are listed in the ListBox in an

ascending order from one to the entered number. And when click the button "أعداد زوجية" the Even numbers are listed in the ListBox in an ascending order from one to the entered number.

- 1) Press Double click on the button "اعداد فردية" then write the following statements by using (For...Next) or (Do While...Loop) to display the *Odd* numbers in the ListBox

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim N, i As Integer
    N = TextBox1.Text
    ListBox1.Items.Clear() REM لافراغ قائمة النص
    For i = 1 To N Step 2
        ListBox1.Items.Add(i) REM لإضافة قيمة لقائمة العناصر
    Next
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim N, i As Integer
    N = TextBox1.Text
    ListBox1.Items.Clear() REM لافراغ قائمة النص
    i = 1
    Do While i <= N
        ListBox1.Items.Add(i)
        i = i + 2
    Loop
End Sub
```

2) Press Double click on the button "اعداد زوجية" then write the following statements by using (**For...Next**) or (**Do While...Loop**) to display the *Even* numbers in the ListBox

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim N, i As Integer
    N = TextBox1.Text
    ListBox1.Items.Clear() REM لافراغ قائمة النص

    For i = 2 To N Step 2
        ListBox1.Items.Add(i) REM لإضافة قيمة لقائمة العناصر
    Next

End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim N, i As Integer
    N = TextBox1.Text
    ListBox1.Items.Clear() REM لافراغ قائمة النص

    i = 2
    Do While i <= N
        ListBox1.Items.Add(i) REM لإضافة قيمة لقائمة العناصر
        i = i + 2
    Loop

End Sub
```

Exercise (3-6) :

Design a project to receives a positive number in the TextBox, when click the button "مجموع الأعداد الفردية" the **Sum** of Odd numbers (from one to the entered number) are displayed in the tool "Label3". And when click the button "مجموع الأعداد الزوجية" the **Sum of Even** numbers (from one to the entered number) are displayed in the tool "Label3".

- 1) Press Double click on the button "مجموع الاعداد الفردية" then write the following statements by using (**For...Next**) or (**Do While...Loop**) to display the *Sum of Odd* numbers in the Label3

```
Private Sub Button1_Click(ByVal
    Dim N, i, Sum As Integer
    N = TextBox1.Text

    For i = 1 To N Step 2
        Sum = Sum + i
    Next

    Label3.Text = Sum
End Sub
```

```
Private Sub Button1_Click(ByVal
    Dim N, i, Sum As Integer
    N = TextBox1.Text

    i = 1
    Do While i <= N
        Sum = Sum + i
        i = i + 2
    Loop

    Label3.Text = Sum
End Sub
```

- 2) Press Double click on the button "مجموع الاعداد الزوجية" then write the following statements by using (**For...Next**) or (**Do While...Loop**) to display the *Sum of Odd* numbers in the Label3

```
Private Sub Button2_Click(ByVal
    Dim N, i, Sum As Integer
    N = TextBox1.Text

    i = 2
    Do While i <= N
        Sum = Sum + i
        i = i + 2
    Loop
    Label3.Text = Sum
End Sub
```

```
Private Sub Button2_Click(ByVal
    Dim N, i, Sum As Integer
    N = TextBox1.Text

    For i = 2 To N Step 2
        Sum = Sum + i
    Next
    Label3.Text = Sum
End Sub
```

الإجراءات : Procedures

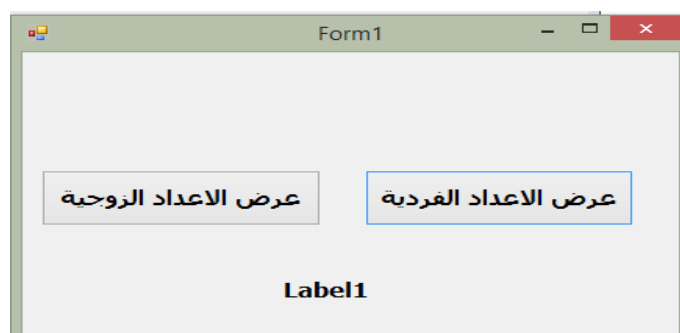
- ❖ When you add a new form window, a new class is created as Form1.
- ❖ Within the scope of this class we declare:
 - 1 - Event procedures.
 - 2 - Variables.
 - 3- Constants.

```
Public Class Form1
    Dim total As Integer
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        ' عرض الاعداد الفردية من 1 الى 10
        Dim i As Integer
        Label1.Text = ""
        For i = 1 To 10 Step 2
            Label1.Text = Label1.Text & " " & i
        Next
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        ' عرض الاعداد الزوجية من 1 الى 10
        Dim i As Integer
        Label1.Text = ""
        For i = 2 To 10 Step 2
            Label1.Text = Label1.Text & " " & i
        Next
    End Sub
End Class
```

In the previous code, we can notice the declaration of the following

1. "**Class**" is under the name (Form1):
2. "**Variables**" are under the names (total , i):
3. "**Event procedures**" are under the name of (Button1_Click, Button2_Click:)



Procedure

A set of commands and instructions under a name, can be recalled by that name, to implement them, and create a (Sub) if we have a set of commands that are frequently used in more than one place in the class

NOTICE

- ♣ Procedures must be called by their names.
- ♣ Calling a procedure causes the program to execute procedure's statements or code.
- ♣ Variables and constants, that can be declared either within the scope of the (Event procedures) or within the (Class).
- ♣ You can declare the Procedures, this declaration is done only once, but you recall the procedures many times from anywhere in your program.

The types of Procedures

There are **two** types of procedures in Visual Basic .NET:

1. **Sub** procedures (اجراء فرعى)
 2. **Functions** (الدالة) procedures.
- ❖ Sub procedures do **not return** a value (لا يعود بقيمة), while Functions **return** a value.

Declaration of a Sub Procedure:

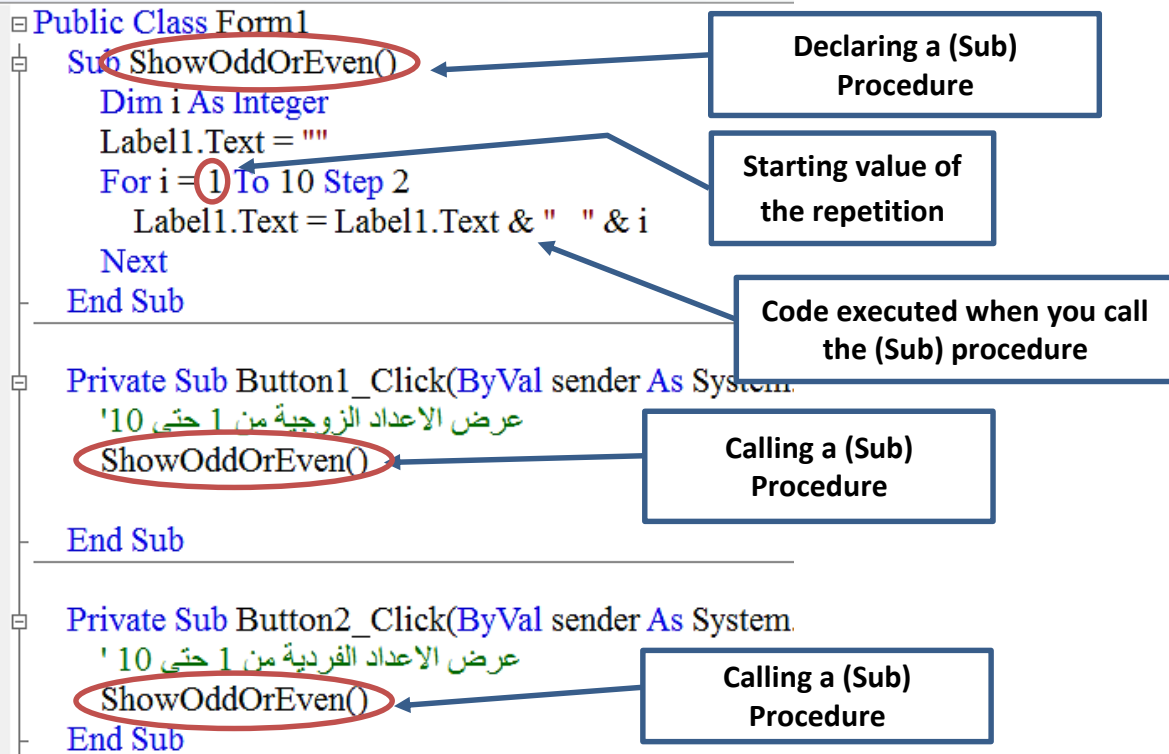
You can declare a Sub procedure in a **class**; if we had a code that will be **repeated** in more than one place in this Class; as well as for the organization of this code, and so it will be easy to read and understand. And then modify it if necessary.

Where:

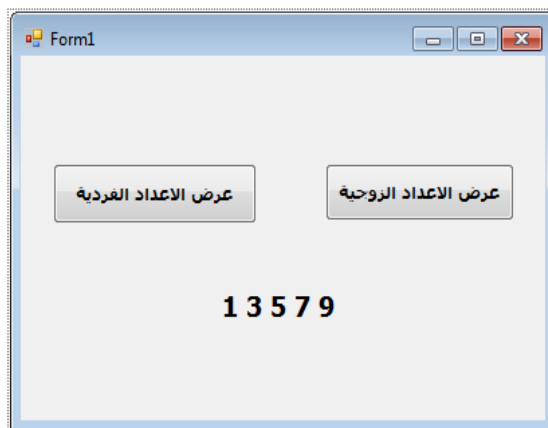
- 1) "**Name**" reflects the name of the procedure.
- 2) "**Parameters**" reflect the values that were used inside the procedure code that are used on recalling the procedure.
- 3) "**Code**" is a set of orders and instructions carried out on recalling the procedure (Sub).

Declaration Syntax :

```
Sub Name (Parameters)
    Code
End Sub
```


Exercise (3-7) :

- ✚ We declared a procedure called (**ShowOddOrEven**) that contains the repeated code.
- ✚ The procedure is executed by typing its name in any other procedure within the class, such as typing its name in each of the event procedures (`Button1_Click`) and (`Button2_Click`).

**NOTICE :**

When testing the previous program, we find that both buttons, the (عرض الاعداد الزوجية) button and the (عرض الاعداد الفردية) button; when clicking any of them, give the same result (1 3 5 7 9), because the starting value of the iteration (repetition) in both procedure has the same value 1.

Declaring and using Parameters :

Exercise (3-8) :

- ❖ To solve the previous problem, the procedure (ShowOddOrEven) must receive the values (1) or (2) on recalling it. this value is used to specify whether the odd numbers will be displayed or the even numbers will be displayed. So we add the variable (Start) that will be called later.

```
Public Class Form2
    Sub ShowOddOrEven(ByVal Start As Integer)
        Dim i As Integer
        Label1.Text = ""
        For i = Start To 10 Step 2
            Label1.Text = Label1.Text & " " & i
        Next
    End Sub
```

Declaring a Parameter without "Dim"

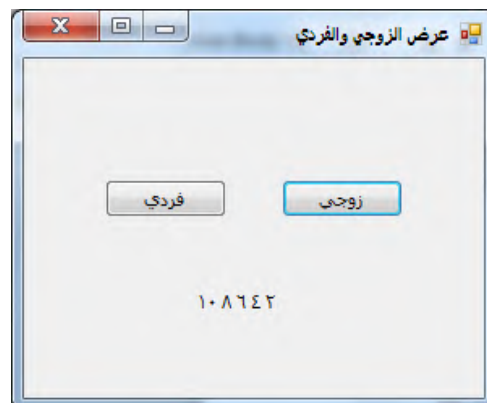
Using a Parameter

- ❖ In figure, a sub Procedure of the name (**ShowOddOrEven**) has been declared and a *Parameter* named (**Start**) has been also declared. And used in the code to specify the starting value of the iteration (repetition). accordingly it displays odd or even numbers

```
Private Sub Button1_Click(ByVal sender As Sy
    REM عرض الاعداد الزوجية من 2 حتى 10
    ShowOddOrEven(2)
End Sub

Private Sub Button2_Click(ByVal sender As Sy
    REM عرض الاعداد الفردية من 1 حتى 10
    ShowOddOrEven(1)
End Sub
```

Setting an Argument value



NOTICE

You can develop the sub Procedure of name (ShowOddOrEven) that receives the start value and end value in the iteration (repetition)

```
Public Class Form1
    Sub ShowOddOrEven(ByVal Start As Integer, ByVal LastValue As Integer)
        Dim i As Integer
        TextBox3.Text = ""
        For i = Start To LastValue Step 2
            TextBox3.Text = TextBox3.Text & " " & i
        Next
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As Syst
        REM عرض الاعداد الزوجية من 2 حتى 10
        ShowOddOrEven(2, 10)

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As Syst
        ' عرض الاعداد الفردية من 1 حتى 10
        ShowOddOrEven(1, 10)

    End Sub
```

Declaring the parameters

Using the parameters

Calling the procedure (ShowOddOrEven)

Declaration of a Function :

Function is a set of commands under a particular name that should express its task. It is applied to **Parameters** and **Returns a value**

Declaration Syntax :

```
Function Function Name (Parameters) As Data Type
    Code
    Return Value
End Function
```

Where:

- 1) "**Name**" expresses the name of the function.
- 2) "**Datatype**" identifies the type of the returned value of the function.
- 3) "**Parameters**" represents the parameters that will be used in the code.

- 4) "Code" is a set of commands and instructions that will be executed on calling the Function.
- 5) "Value" is the returned value by the function.

Exercise (3-9): Calculate the sum of two numbers:

- 1) Design a Form window as shown in figure.
- 2) Open the Code Window, press (F7) then type this Code

Declared the (Sum) Function

The Data type of the value returned by the function

```
Public Class Form1
    Function Sum(ByVal First As Single, ByVal Second As Single) As Single

        Dim total As Single

        total = First + Second

        Return total

    End Function

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim x As Single = TextBox1.Text
        Dim y As Single = TextBox2.Text

        Label4.Text = Sum(x, y)

    End Sub
End Class
```

Declared of the parameters used in code

Calling the function

The code window where the function (sum) is declared and calling the (sum) function by Create an event procedure for Button1,

NOTICE :

- + We declared the (**Sum**) Function; of Data Type (**Single**) that receives two values from the function variables (**First** and **Second**).
- + A variable named **total** have been declared of type (**Single**).
- + The sum of the two values **First** and **Second** will be assigned to the variable **total**.
- + After the function executes, **the return statement** sends a value back to the statement and it is assigned **to total**.
- + The **value total** is stored in the name of **the function (Sum)** as will be seen when it is called.

Remember :

- 1) **Variables:** We can assign values to Variables; during the declaration and the execution of the Program instructions, as well as using these values stored.
- 2) **Constants:** We can assign values to Constants; during the declaration only, as well as using these values stored.
- 3) **Functions:** We cannot assign values to Functions, but function can be called and returns a value in the light of the values assigned to the function, as well as using this value stored.

NOTICE:

- + It is preferred when naming Functions; give names related to their functionality.
- + You can:
 1. Declare a Function.
 2. Determine its Parameters.
 3. Specify the Function type.
 4. Write Code within this Function.
 5. Return a value using the Return statement.

NOTICE :

Sub Procedure	Function
begins with a (Sub) statement	begins with a (Function) statement
do not return a value	return a value
are not used in the assignment statement	All functions must be written on the right side of the assignment to get the results of these functions
Parameters given to sub Procedure can be (an abstract value or a variable or a constant or a function)	Also , Parameters given to Function can be (an abstract value or a variable or a constant or a function)
The Event Procedure is considered a (Sub) Procedure.	MsgBox () considered a function
It has not data type <code>Sub ShowOddOrEven(ByVal Start As Integer)</code>	It has data type <code>Function Factorial(ByVal Number As Integer) As Integer</code>

Questions**(Q): Answer the questions with the help of the following code:**

- The code is executed when you press on control tool.....
- "Dim" is used to declare (variable – constant) with type.....
- The variable name used in the iterative loop is..... :
- The starting value of the iterative loop is , the end value is and the value of increment is.....

```
Private Sub Button1_Click(ByVal  
    Dim M As Integer  
    For M = 1 To 3  
        MsgBox (M)  
    Next  
End Sub
```


- E. Implementation of the iterative loop stops when the value of variable M reaches.. .. .
- F. The code that is repeated is.

(2): Answer the questions with the help of the following code:

```
Private Sub But_Repeat_Click (ByVal sender As System.Object,
    Dim m As Integer
    Me.Label1.Text = ""
    For m = 5 To 9 Step 2
        Me.Label1.Text = Me.Label1.Text & m & vbCrLf
    Next m
    ..... (The required in line number 7)
    MsgBox ("انتهى البرنامج")
End Sub
```

- (A) The purpose of the code is:
.....
- (B) the code is executed when the event occurs on the control tool
- (C) to declare the variable m, the command is used.
- (D) The loop statement used is
- (E) The code to be repeated is
- (F) The purpose of the use of concatenation operator & in a the statement
(Me.Label1.Text = Me.Label1.Text & m) is. :
- (G) Type the necessary code to display the final value of the variable M after the
execution of the iterative loop in a message box:
.....

(3): Answer the questions with the help of the following code

```
Dim n, product As Integer
Dim str As String
Me.TextBox1.Text = ""
For n = 1 To 12
    Str = 3 & " × " & n & " = "
    product = 3 * n
    Me. TextBox1.Text = Me. TextBox1.Text & str &
        product & vbCrLf
Next n
End Sub
```

- A- The purpose of the code is
- B. The purpose of the code (Dim str As String) is to declare a string variable named str. (True - false)
- C- The purpose of the code (product = 3 * n) is assigning the result of multiplying 3 by the variable n to the variable product. (True - false)
- D- The purpose of the code (product = 3 * n) is assigning the result of multiplying 3 by the variable product to the variable n. (True - false)
- E. The purpose of the code:

Me.TextBox1.Text = Me.TextBox1.Text & str & product & vbCrLf

is assigning the value of the string variable "str" and the value of the variable "product" as a value for the property "text" for TextBox1. (True - false)

- F. The purpose of the code part "vbCrLf" is to transition to a new line. (True false)

(4): The following code is for typing a multiplication table of number 4 from 1 to 12.

Required: Modify the code to have a multiplication table of (7) so that the result will be in a TextBox.

$$7 \times 5 = 35$$

$$7 \times 7 = 49$$

$$7 \times 9 = 63$$

$$7 \times 11 = 77$$

```
Dim n, product As Integer
Dim str As String
Me.TextBox1.Text = ""
For n = 1 To 12
    Str = 4 & " × " & n & " = "
    product = 4 * n
    Me. TextBox1.Text = Me. TextBox1.Text & str &
        product & vbCrLf
Next n
End sub
```


(5): The purpose of the following code is to type a multiplication table of (9) by the numbers from 1 to 10.

Required: Correct the four errors in the code, until we get the correct result of the code execution in the table.

```

Dim n, product As String
Dim str As String
Me.TextBox1.Text = ""
For n = 1 To 10 Step -1
    Str = 9 & " × " & n & " = "
    product = 9 + n
    Me. TextBox1.Text = Me. TextBox1.Text & str &
        product & vbCrLf
Next str
End Sub

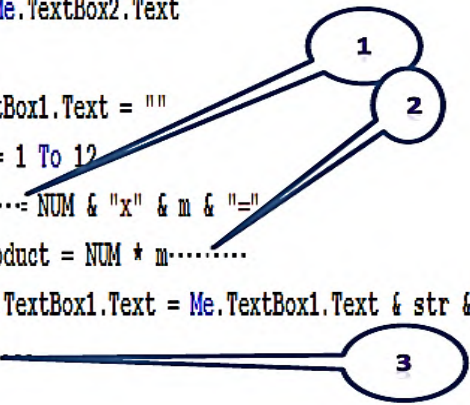

```



No.	Wrong code	The code after correction
1
2
3
4

(6): The following table contains the code and the form window to print a multiplication table for any number from 1 to 12

Required: In the following table, complete using the necessary code to get the correct output after running the program .

الكود	نافذة تشغيل البرنامج
<pre> Dim m, p..... NUM As Integer Dim str As String NUM = Me.TextBox2.Text Me.TextBox1.Text = "" For m = 1 To 12 = NUM & "x" & m & "=" product = NUM * m..... Me.TextBox1.Text = Me.TextBox1.Text & str & product & vbCrLf N..... </pre> 	

(7) With the help of the code, answer the following questions:


```
For I = 1 To B Step C
```

```
    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf
```

```
Next
```

- A- The purpose of the code:.....
- B- The name of the counter variable:.....
- C- The iterative loop begins with the value.....
- D- The iterative loop ends with the value.....
- E- the value of the increment of counter.....
- F- The purpose of vbCrLf is.....

(8) The following code is used to input a positive number, and when you press the "odd numbers" button, odd numbers from 1 to the positive number that has been entered is typed, and if you press, "Even numbers" button, odd numbers from 1 to the positive number that has been entered is typed.

الكود	نافذة تشغيل البرنامج
<pre>Dim N, i As Integer N = TextBox1.Text ListBox1.Items.Clear() i = 1 Do While i <= N ListBox1.Items.Add(i) i = i + 2 Loop</pre>	

Required:

- (1) The **loop statement** in the program is..... :
- (2) The purpose of the code (I = I + 2) in the line before the last is..... :
- (3) The purpose of the Loop is:
- (4) Select the right choice to determine the nature of each part of the line of code components:

ListBox1.Items.Clear()

○ أداة تحكم

○ متغير

○ ثابت

○ خاصية

○ وسيلة

○ خاصية

(9) The purpose of the following code is entering a positive number, then the sum of odd numbers is displayed in text box.

(A) The purpose of the code N = TextBox1.Text is

(B) The Loop statement used in the code is :

(C) The loop statement will be implemented as long as

(D) The sum of odd numbers is displayed in the text box when we get to a number larger than the positive one that has been entered in the control tool and that was assigned to the variable.....

```
Dim N, i, sum As Integer
N = TextBox1.Text

i = 1
Do While i <= N
    sum = sum + i
    i = i + 2
Loop

Label3.Text = sum
```

(10): Tick (✓) in front of the correct statement and a sign (✗) in front of the wrong one for each of the following phrase:

No	Question	Answer
1	The procedure is a set of commands and instructions that are repeated for specified number of times.	()
2	The procedure is a set of commands and instructions under a certain name, and when you recall this name, these commands and instruction are implemented.	()
3	The purpose of the use of procedures is repeating typing a specific code several times in the program.	()
4	When we have a specific code that we want to be replicated in more than one place in the class, we use the " Function "	()
5	The group of Commands and instructions that are placed under a name, when we implement them, they return a value. we call this action " procedure ".	()
6	The group of Commands and instructions that are placed under a name, when we implement them, they return a value. We call this action " Function ".	()
7	When we have a specific code that we want to be replicated in more than one place in the class, we use the " Procedure ".	()
8	Parameters are used to receive values from outside the procedure on recalling it.	()
9	When you recall a procedure with the name Taxes(0.05), the value between the brackets is called Argument .	()
10	When you call a procedure with the name Taxes(0.05), the word taxes is called Argument .	()
11	The declaration of a function starts with (Sub) and ends with (End Sub).	()
12	The declaration of a function starts with (Function) and ends with (End Function).	()
13	We resort to the use of the Function if our code results in a value we need.	()
14	We resort to the use of the Procedure if our code results in a value we need.	()

15	The Function is a set of commands and instructions with a specific name that can take Parameters , and return a parameter .	()
16	The Function is a set of commands and instructions with a specific name that can take values, and return a value .	()
17	The demerit of the language of VB.Net is that it allows the programmer to declare other functions and procedures prepared by him.	()

(11) With the help of the following code, complete the following table:

```
Sub ShowOddOrEven(ByVal Start As Integer)
    Dim i As Integer
    Label1.Text = ""
    For i = Start To 10 Step 2
        Label1.Text = Label1.Text & " " & i
    Next
End Sub
```

No	Question	Answer
1	Procedure Name
2	Parameter was declared with the name ... and type
3	The iterative loop starts from the value
4	The increment value of the iterative loop equals
5	When calling the procedure to execute the code starting from the value
5	Feedback from the function value.

(12) With the help of the code, complete the following table:

Function XXX (ByVal YYY As Integer, ByVal ZZZ As Integer) As Single

Code

Return RRR

End Function

No	Required	Answer
1	Function name
2	Type of returned value of the function
3	Parameter value that will be used in the code
4	Returned value of the function

Chapter Four

Cyber Bullying

التعدي الإلكتروني

Through the internet we can learn – entertain – talk – communicate ... but there are many risks that we can be exposed to including:

- Getting wrong information.
- Falling prey to some of the aggressors across modes of electronic communication.
- Violation of privacy.
- Identity theft.
- Getting our account stolen (on the social networking sites like Facebook or email).
- Subjecting our system to the risk of infection by viruses or spyware, or software piracy and others.

Cyber Bullying definition:

Cyber bullying is a deliberate aggressive (عدوان متعمد) behavior from one person to another through electronic modes of communication.

Firstly: The forms of cyber bullying:

1. Harassment (المضايقة – التحرش)
2. Annoyance (التخفى)
3. Embarrassment (الإحراج)
4. Intimidation (التخويف)
5. Threat (التهديد)
6. Blackmailing
7.etc....

Secondly: Electronic media used in Bullying:

Electronic media is a technology used by the electronic aggressor, and they are various including the following

1. E-Mail (البريد الإلكتروني)
2. Forums (المنتديات)
3. Instant Message (الرسائل الفورية)
4. Blogger (التدوين الإلكتروني)
5. Facebook.

Thirdly: Forms of Cyber Bullying:

Forms of cyber bullying include:

1. **Anonymity:**
The use of pseudonyms (aliases) to hide e-aggressor's identity for impunity.
2. **Harassment:**
It is aggressive messages directed against one or more persons.
3. **Cyber stalking:**
It is a form of electronic harassment where the aggressor frequently traces and chases a particular person in all electronic media.
4. **Flaming:**
It is a publication of hostile and vulgar words against one or more through a media and electronic communication
5. **Outing:**
It is a dissemination of information about a specific person or more abusively.
6. **Exclusion:**
It is to ignore one or more persons through the electronic media.
7. **Cyber threats:**
It is an email or e-message carrying a threat and intimidation to one or more persons.

Fourthly: How to protect yourself from Cyber Bullying:

By following the instructions below:

1. Do not share a password with anyone.
2. Make a password that is difficult to predict.
3. Don't publish (post) (نشر) any private data.
4. Avoid deleting Cyber bullying messages.
5. Don't interview (مقابلة) anyone you know via the internet.
6. Be careful! Don't send messages when you are angry.
7. Inform (إطلاع) your parents with what annoy (يضايقك) you when you use the Internet.

8. The download of software from the Internet should be done under the supervision (إشراف) of your teacher or your parents.

Questions

First: State whether the following statements are (true) or (false):

No	Question	Answer
1	Cyber bullying is a deliberately aggressive behavior, using electronic media for harassment, annoyance, disturbance, intimidation or threatening others.	()
2	Cyber bullying is done through electronic means, such as social networking sites	()
3	Stealthy-mail is considered a form of Cyber bullying.	()
4	Harassment and the threat are of the most important electronic means used in Cyber bullying	()
5	Harassment and blackmailing are forms cyber bullying	()
6	Stealing the person's account in the social networking sites or email is one of the risks that we may be exposed to through such media	()
7	Social networking sites help to meet new people you like to see to develop social relationships	()
8	In line with the rules of safe use. you should put an easy password passage ,for your private e-mail, in order to be able to remember,	()
9	E-exception means following a particular person in all means of electronic communication.	()
10	Electronic prosecution is intended to send an e-mail carrying a threat and holiday for one or more persons.	()

Second: Complete the following table explaining your opinion of each of the following words:

No	Question	Answer
1	Putting a password that's easy to deduce	()
2	Someone published his real name, address and telephone number through the electronic media.	()
3	Download any available program to you on the internet	()
4	React angrily to cyber bullying you may be exposed to on the internet.	()
5	Delete all messages of threat on networking sites or e-mail	()

Third: Everyday situations:

- 1) Someone subscribed to one of social networking sites. Whenever he contacted a member or tried to conduct an immediate conversation, he noticed that no one responded. What happened is considered a form of..... and it's called..... What can you do to face this behavior
- 2) You subscribed to one of social networking sites and you were surprised that someone was sarcastically speaking about Egypt and talking inappropriately about its figures.
- Select four positive actions by which you can respond to him:

.....



تطبيق مذكرات جاهزة للطباعة



حمل التطبيق على موبايلك الأندرويد أو الأيفون

موقع مذكرات جاهزة للطباعة - www.cryp2day.com